

# Quarkus JMS Communication

JEE Microservices  
@ CGS IT – 2023

v.1.0.3



QUARKUS

# Inhalt

- Quarkus – Voraussetzungen - Prerequisites
- Maven Quickstart: Create by hand
- Quarkus Quickstart: create via IntelliJ
- Quickstart : launch new app
- Quarkus – Generated Output
- Quarkus Maven plugin
- Start Quarkus App
- Quarkus – Rest API
- Add to git
- Import into IntelliJ
- IntelliJ Project
- Maven Settings in IntelliJ
- IntelliJ: Java Settings for Java Project
- Run Quarkus from IntelliJ maven
- Quarkus Development Mode
- IntelliJ - Quarkus – Attach Debugger
- Quarkus - Testing

# Quarkus JMS

- AMQPP oder Artemis JMS Extension

```
<dependency>  
  <groupId>org.amqphub.quarkus</groupId>  
  <artifactId>quarkus-qpj-jms</artifactId>  
</dependency>
```

```
<dependency>  
  <groupId>io.quarkiverse.artemis</groupId>  
  <artifactId>quarkus-artemis-jms</artifactId>  
</dependency>
```

# Quarkus JMS – Starting JMS Broker

# Quarkus JMS – Sender Example

```
@ApplicationScoped

public class PriceProducer implements Runnable {

    @Inject
    ConnectionFactory connectionFactory;

    private final ScheduledExecutorService scheduler = Executors.newSingleThreadScheduledExecutor();

    void onStart(@Observes StartupEvent ev) {
        scheduler.scheduleWithFixedDelay(this, 0L, 5L, TimeUnit.SECONDS);
    }

    void onStop(@Observes ShutdownEvent ev) {
        scheduler.shutdown();    }

    @Override
    public void run() {
        try (JMSContext context = connectionFactory.createContext(JMSContext.AUTO_ACKNOWLEDGE)) {
            context.createProducer().send(context.createQueue("prices"), Integer.toString(random.nextInt(100)));
        } } }
```

# Quarkus JMS – Consumer Service

```
@Override
public void run() {
    try (JMSContext context = connectionFactory.createContext(JMSContext.AUTO_ACKNOWLEDGE)) {
        JMSConsumer consumer = context.createConsumer(context.createQueue("prices"));
        while (true) {
            Message message = consumer.receive();
            if (message == null) return;
            lastPrice = message.getBody(String.class);
        }
    } catch (JMSException e) {
        throw new RuntimeException(e);
    }
}
```

Danke für Ihre Aufmerksamkeit